Coding Lab 4: Global Alignment – NeedlemanWunsch Algorithm

Objective

Implement the global (Needleman-Wunsch) alignment algorithm with simple scoring: match, mismatch, linear gap penalty.

Background

Sequence alignment allows comparison of biological sequences, locating similarities, indels, and evolutionary relationships.

Tasks

1. Write a function that builds the scoring matrix for two sequences and returns it.
2. Implement the traceback to get aligned sequences.
3. Test your implementation on short sample sequences.
   - For example, x = CGATCCTGT, y = CATCGCCTT
4. Try to create a function that formats the alignment like this:

```
C G A T - - C C T G T
|   | |     | | |   |
C - A T C G C C T - T
```

Hints

- Gap should be ≤ 0 and match $>$ mismatch.
- `numpy` defaults to `float`, forcing `int` might be a good idea.
- While computing the F-matrix consider creating an additional pointer matrix which keeps track of the *direction*. With this matrix you can accelerate traceback.
- Use explicit determinism when choosing directions.

Optional extensions

- Implement affine gap penalty (gap open + gap extend).
- Score matrices for proteins (e.g. BLOSUM62).
- Local alignment (Smith–Waterman).
- Hirschberg's algorithm (divide-and-conquer) for $O(n)$ complexity